VAASAN AMMATTIKORKEAKOULU
VASA YRKESHÖGSKOLA
UNIVERSITY OF APPLIED SCIENCES

Hu Hai

Automatic Repeat-Request Courseware

Technology and Communication

2013

VAASAN AMMATTIKOREAKOULU
UNIVERSITY OF APPLIED SCIENCES
Degree Programme of Information Technology

ABSTRACT

Author	Hu Hai
Title	Automatic Repeat-Request Courseware
Year	2013
Language	English
Pages	44
Name of Supervisor	Gao Chao

An Automatic Repeat Request (ARQ) courseware is demanded from the teacher for the course IIA0502 “Basics of Computer Networks”. A java applet is the target of this design. The user interface must give the user freedom to select necessary parameters, simulate Stop-and-wait, Go-Back-N, Selective Repeat ARQ data transfer process, show the result, allow students have more intuitive understanding of the ARQ theory.

Although this ARQ simulation courseware is not perfect running, some bugs are exist, still those simulation results could meet the theory.

ABBREVIATIONS

ACK Acknowledgement

ARQ Automatic Repeat-reQuest

BER Bit Error Rate

FER Frame Error Rate

NAK (NACK) Negative-Acknowledgement

CONTENTS

ABSTRACT
ABBREVIATIONS
LIST OF FIGURES AND TABLES
1 INTRODUCTION 7
1.1 Outline of the Thesis 7
2 ARQ THEORY 8
2.1 Flow Control 9
2.2 Error Control 9
2.3 Stop-and-wait ARQ 9
2.4 Go-Back-N ARQ 16
2.5 Selective-Repeat ARQ 22
3 JAVA SOFTWARE DESIGN 25
3.1 Use Case 25
3.2 Data Structure 31
4 TESTING AND CALCULATION 33
5 SUMMARY AND CONCLUSION 43
REFERENCES 44

LIST OF FIGURES AND TABLES

Figure 1.	Stop-and-wait ARQ normal operation	p. 11
Figure 2.	Stop-and-wait ARQ lost frame	p. 12
Figure 3.	Stop-and-wait ARQ lost ACK	p. 13
Figure 4.	Stop-and-wait ARQ delayed ACK	p. 14
Figure 5.	Stop-and-wait ARQ bidirectional transmission	p. 15
Figure 6.	Go-Back-N ARQ sender sliding window	p. 17
Figure 7.	Go-Back-N ARQ receiver sliding window	p. 17
Figure 8.	Go-Back-N ARQ control variable	p. 18
Figure 9.	Go-Back-N ARQ normal operation	p. 19
Figure 10.	Go-Back-N ARQ lost frame	p. 20
Figure 11.	Go-Back-N ARQ sender window size	p. 21
Figure 12.	Selective Repeat ARQ receiver slide window	p. 22
Figure 13.	Selective Repeat ARQ lost frame	p. 23
Figure 14.	Selective Repeat ARQ sender window size	p. 24
Figure 15.	Use case	p. 25
Figure 16.	Protocol choosing menu 1	p. 26
Figure 17.	Protocol choosing menu 2	p. 27
Figure 18.	Stop-and-wait ARQ user interface	p. 28
Figure 19.	Go-Back-N ARQ user interface	p. 29
Figure 20.	Selective Repeat ARQ user interface	p. 30
Figure 21.	Data structure	p. 31
Figure 22.	Stop-and-wait ARQ parameter calculation	p. 33
Figure 23.	Stop-and-wait ARQ in progress	p. 34
Figure 24.	Stop-and-wait ARQ result	p. 35
Figure 25.	Go-Back-N ARQ parameter calculation	p. 37

Figure 26. Go-Back-N ARQ in progress p. 38

Figure 27. Go-Back-N ARQ result p. 39

Figure 28. Selective Repeat ARQ parameter calculation p. 40

Figure 29. Selective Repeat ARQ in progress p. 41

Figure 30. Selective Repeat ARQ result p. 42

Figure 31. Three ARQ protocols comparing table p. 43

1 INTRODUCTION

In the course “Basics of Computer Networks”, An Automatic Request (ARQ) courseware is required. By using Java applet, the courseware can be simply run as an html file. Users can choose three different ARQ schemes (Stop-and-wait, Go-Back-N, Selective Repeat), and choose parameters freely, like frame size, total number of frames to be transmitted, link length (distance between the sender and receiver), link quality in BER (bit error rate), sliding window size (applied if Go-Back-N or Selective Repeat is selected). After the simulation start, the users can see following features: a universal time ticking, timeline at the sender and receiver, frames (showing frames numbers) passing through the link, ACK/NAK travel back, sliding window situation (for Go-Back-N and Selective Repeat ARQ), timer situation. After simulation, the program will show the results: number of erroneous data frames, number of lost frames, number of erroneous ACK/NAK frames, number of lost ACK/NAK frames, number of retransmitted data frames and total delivery time.

By using this Java applet simulation courseware, the students could have more intuitive understanding of the ARQ theory. I would explain the telecommunication theory which relates to the courseware and all calculations, software design, testing, simulation results theoretical results comparison, and conclusions.

1.1 Outline of the Thesis

The rest of this thesis is arranged as followed. Chapter 2 talks about ARQ theories. Chapter 3 illustrates the software design in details, including use case and data structure. Chapter 4 tests the courseware and calculates the results. Finally Chapter 5 concludes the work.

2 ARQ THEORY

In the course “Basics of Computer Networks”, data link layer protocols are discussed. The major functionality of data link layer protocols is ensuring error-free communications between a transmitter and a receiver. In a data-oriented communication system, user data is fragmented into a series of small frames and sent out by transmitter, and received and reassembled at the corresponding receiver. During the transmission over a certain type of medium, such as fiber, radio, coaxial cable, twisted pair cable, the frames are vulnerable to noise which are superimposed into the signal, and resulting 1) erroneous bits received, 2) the total frame is lost or cannot be recognized by the receiver. Erroneous bits can be detected by receiver using some redundancy mechanisms, such as parity check, check sum, and cyclic redundancy check (CRC). However, when the above errors occur, a correction mechanism must be applied to correct the errors. Most redundancy methods can only detect errors, they can tell whether a frame is erroneous or not, but cannot locate the errors. Due to the fact that a frame may be totally lost, error correction is achieved by the retransmission of erroneous frames, and this retransmission must be done automatically through the interactions between the transmitter and receiver. Such an automatically achieved method is called Automatic Repeat Request (ARQ). Basically, there are 3 ARQ schemes: Stop-and-wait ARQ, Go-Back-N ARQ and Selective Repeat ARQ.

The data link layer has two main functions, flow control and error control.

2.1 Flow Control

In the data communication, flow control manages the transmitted data between sender and receiver. First, data will be transmit to the receiver, after it is received by receiver, an acknowledgment will be sent back to sender, and then the sender sends the next data. The flow control is responsible for “adjust” the amount of the transmitted data. But the receiver is limited, the incoming data is not allowed to overwhelm the receiver. The receiver needs time to check and process the incoming data and store them before they can be used. But this processing speed is much slower than the data transmission speed. If incoming data almost fills up the receiver buffer memory, the receiver will tell the sender to slow down the transmission speed or even pause.

2.2 Error Control

Error control in data link layer means two things, error detection and error correction. Sometimes, data lost or damaged during the transmission, error control will inform the sender, specify the frames and ask retransmit. This process is called automatic repeat request (ARQ). (In reality, the acknowledgment ACK sent back to sender also could be lost or damaged, but we will not discuss this here. In the demonstration, the error rate of ACK will be assumed to be zero.)

2.3 Stop-and-wait ARQ

Stop-and-wait ARQ is the most basic mechanism among three, and it is the foundation of the other two. In the stop and wait ARQ, frame and ACK are numbered 0 and 1 alternately. Frames 0 sends to receiver, ACK 1 will be sent back to sender; frame 1 goes to receiver, ACK 0 will be back to sender, and so on.

Sending device sends a frame 0 to receiving device, sender keeps a copy of this frame 0. At the same time, there is timer start to timing. When the frame 0 sends to receiver, if this frame correctly received, a positive ACK 1 will be sends back to sender, to tell sender excepting frame 1 transmission. During the transmission, if the frame 0 is lost, receiving device does not receive anything, so it will do nothing. If a damaged frame 0 sends to receiver, receiver will discard this frame 0 automatically and remain silent. But sending device is still waiting the ACK 1 back, after the timer times up, sender assumes the transmission of frame 0 is failed, so it will retransmit the frame 0 to receiver, which is the reason of sender keeps copy of frame 0. There are control variables for sender and receiver, called “S” and “R”. S holds the number of the recently sent frame. R holds the number of excepted frame. The timer duration must be longer than total time of frame sends to receiver and ACK sends back to sender; otherwise it will be an infinite loop.

Normal operation

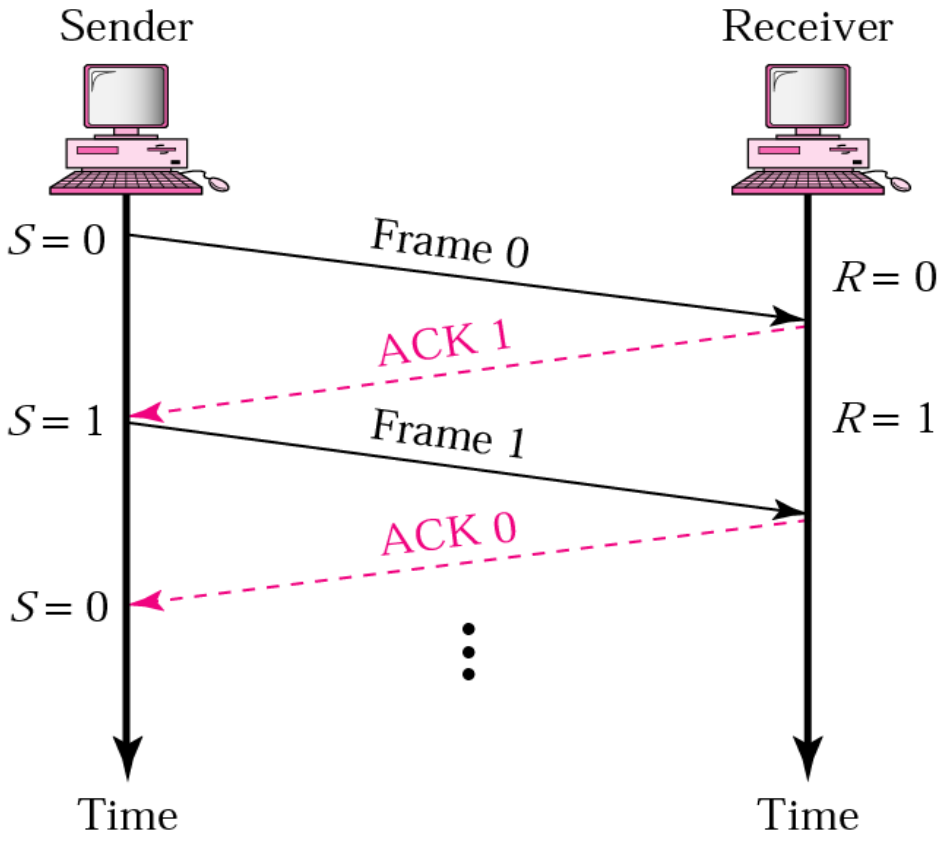

Figure 1. Stop-and-wait ARQ normal operation [1]

Stop-and-wait ARQ, lost frame

As the figure 2 shows, from sender to receiver, frame 1 is lost, but sender is still expecting the ACK 0 back. After timer time out, frame 1 sends again.

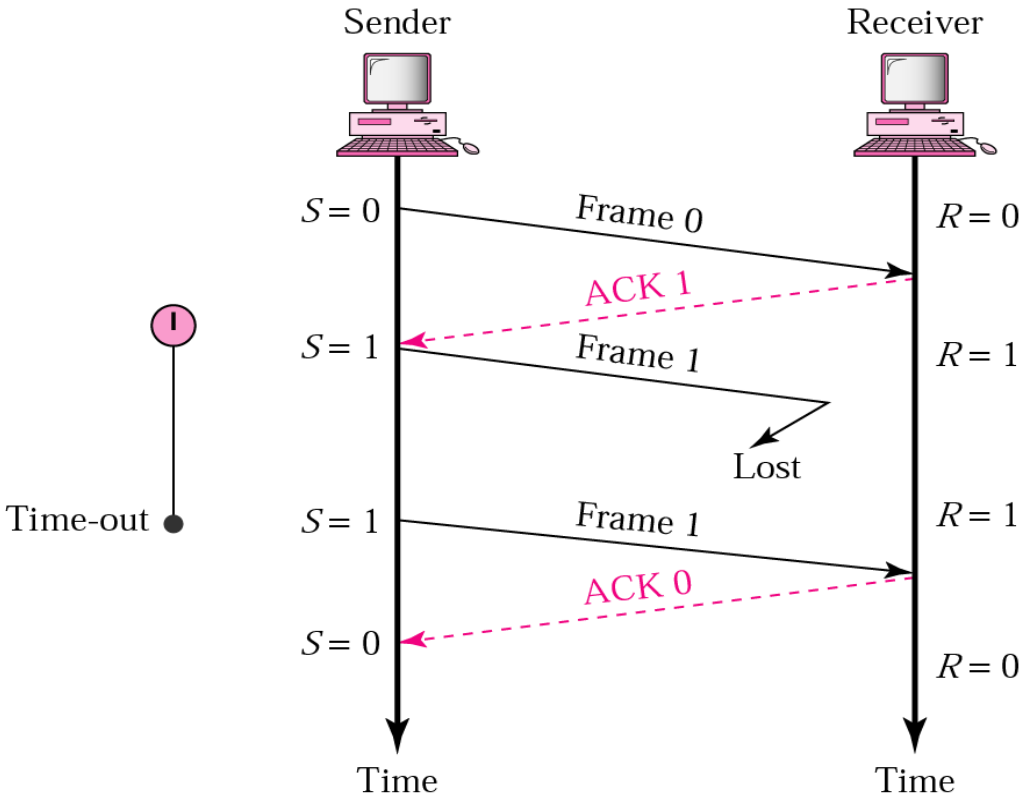

Figure 2. Stop-and-wait ARQ lost frame [1]

Lost or delayed ACK

Stop-and-wait ARQ, lost ACK frame

As the figure 3 shows, when ACK 0 sends back to sender, this frame has lost. So after time out, frame 1 sends again. Receiver side is excepting for frame 0, so frame 1 is discarded.

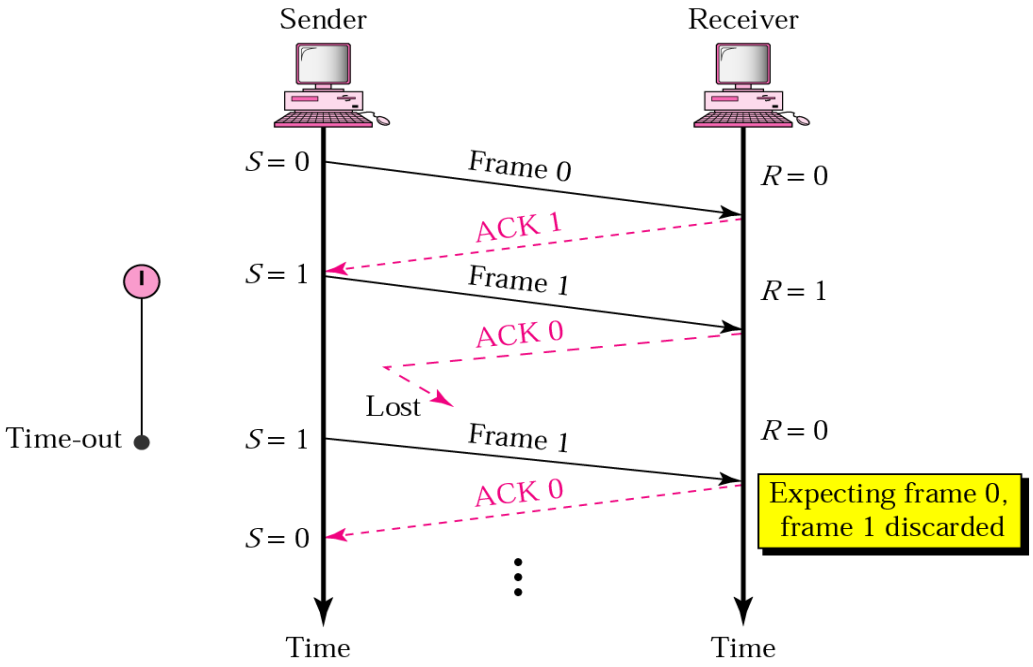

Figure 3. Stop-and-wait ARQ lost ACK [1]

Stop-and-wait ARQ, delayed ACK

Frame 0 sends to receiver, ACK 1 sends back to Sender, but the transmission has delayed. After time out, sender sends frame 0 again; receiver is excepting frame 1. Therefore, the frame 0 is discarded, ACK 1 is discarded. Frame 1 sends again.

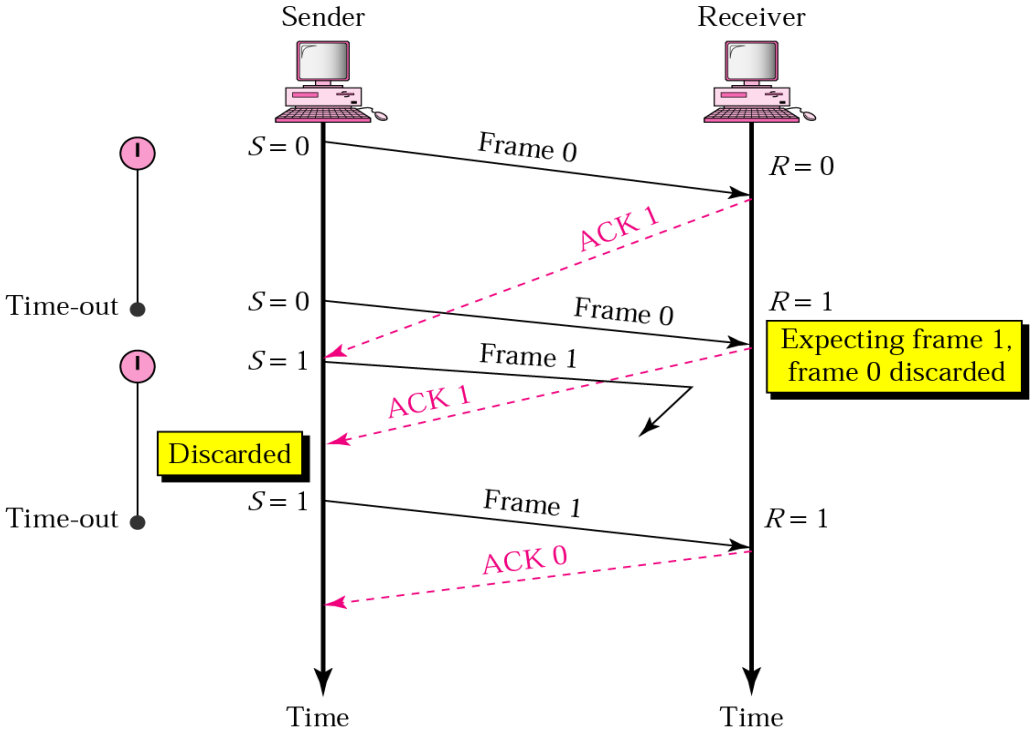

Figure 4. Stop-and-wait ARQ delayed ACK [1]

Bidirectional transmission & Piggybacking

Figure 2, 3 and 4 are unidirectional transmissions. But we could have bidirectional transmission. In that case, both sides are sender and receiver. If the transmissions share the same channel, it is called half-duplex transmission. If they use separate channels, is called full-duplex transmission.

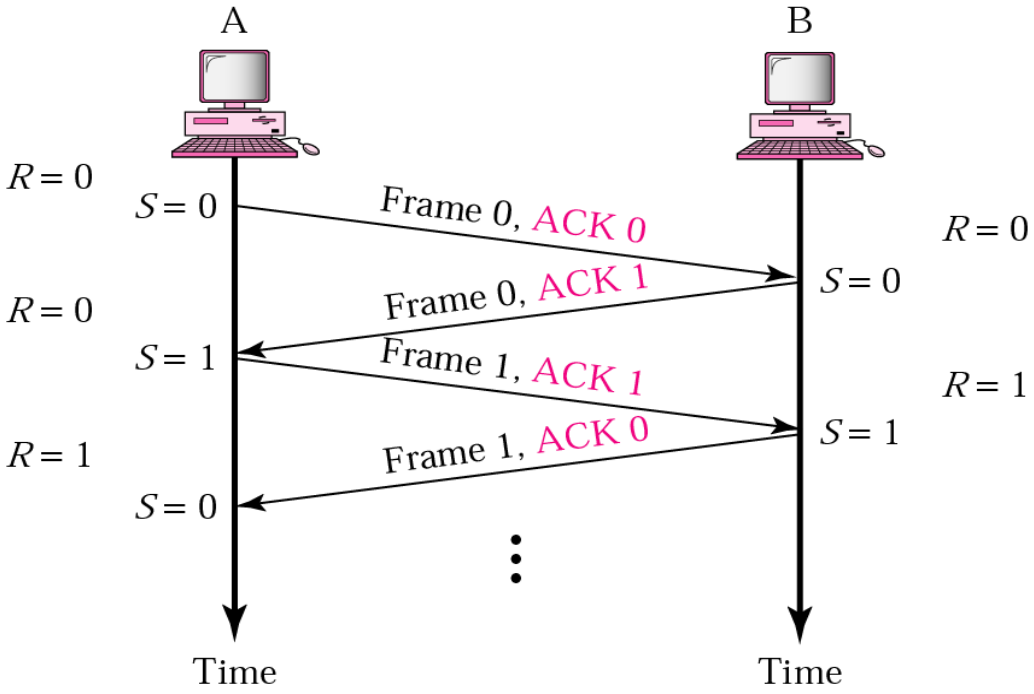

Figure 5. Stop-and-wait ARQ bidirectional transmission [1]

For each frame, minimum size is 64 bytes, the maximum size is 1518 bytes (IEEE 802.1Q) or 1522 bytes (802.3ac).

2.4 Go-Back-N ARQ

If the link quality is very good (assume there is no error frame), for Stop-and-wait ARQ, each time only one frame can be transmitted. But in reality, the link is always noisy, frame lost or damaged happens. The sender and receiver will have nothing to do except waiting. Therefore the Stop-and-wait ARQ is very low efficiency. To improve to this, we can send several frames insides only one. Therefore the Go-Back-N ARQ has been generated.

Sequence Numbers

Because each time, there are several frames waiting for transmission, so we must number them sequentially. We set a limit for the frames. If the header of frame allows m bits for the sequence number, those frames range will be 0 to 2^m-1. If m=2, sequence number will be 0,1,2,3,0,1,2,3… repeat in this way. Unlike Stop-and-wait ARQ's 0,1,0,1…

Sender sliding window

Now, a "group" of frames send to receiver, we need something to hold this "group" until ACK arrived. Next, the concept of "sliding window" is introduced. The window size is fixed which is 2^m-1. Inside this sliding window, there are the copies of the transmission frames. When the correct ACK arrived, sliding window will slid forward.

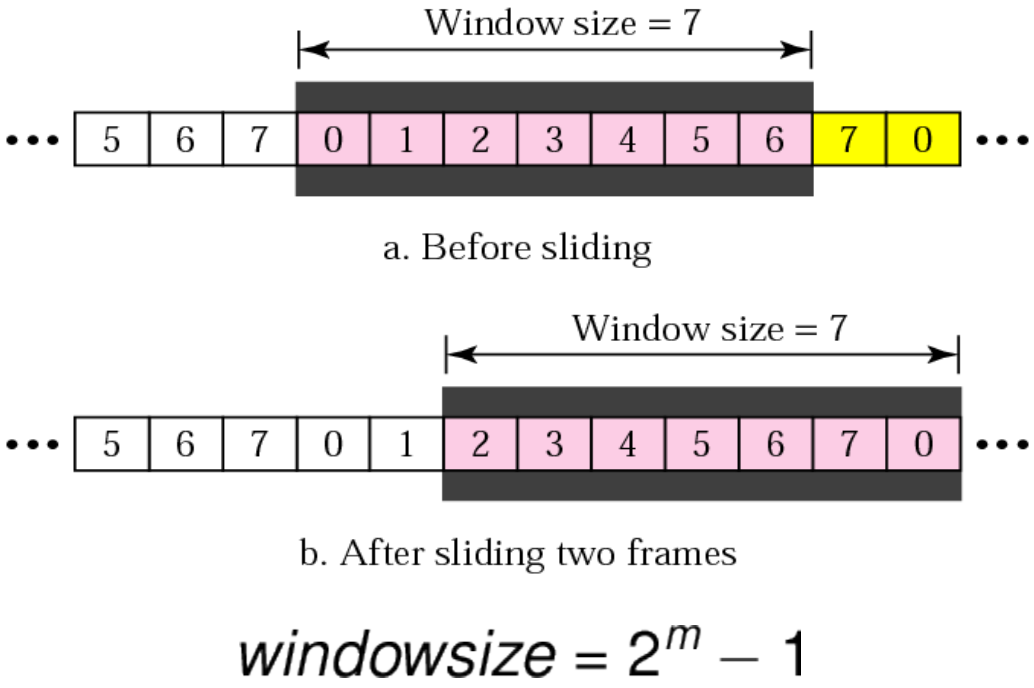

Figure 6. Go-Back-N ARQ sender sliding window [1]

Receiver sliding window

Receiver sliding window in Go-Back-N ARQ is always 1. It's waiting for the correct frame comes in correct order, then sends back the ACK and slide forward. If the frame is lost or damaged, receiver will wait for the resend. Even the rest of the frame is correct, receiver will discard them automatically.

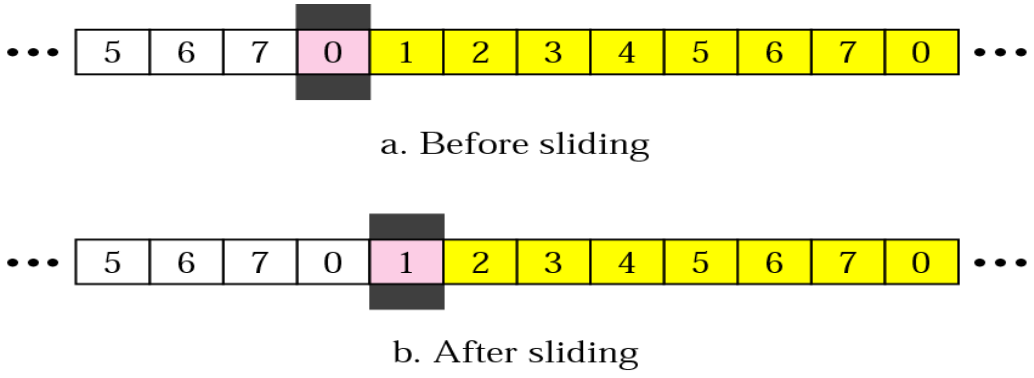

Figure 7. Go-Back-N ARQ receiver sliding window [1]

Control Variables

In the Go-Back-N ARQ, sender's control variables are S, S_F, S_L. But receiver's variable is still R. Slide window size is W. S is the sequence number of latest sent frame, S_F is the sequence number of the first frame in the slide window, S_L is the sequence number of the last frame in the slide window. R is the sequence number of the excepted frame. $W=S_L-S_F+1=2^m-1$. Only when R and sequence number of received frame are matched, frame accept, otherwise discard it.

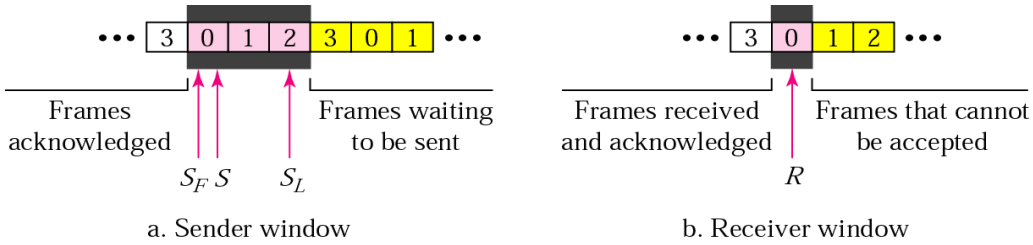

Figure 8. Go-Back-N ARQ control variable [1]

Timer

Inside of slide window, each sent frame has individual timer. The total timer number is equal to the slide window size.

Lost or damaged frame

Receiver will send back an ACK to sender if the correct frame received (right frame in right order). If the frame is lost or damaged, receiver will remain silence. If there is no ACK back not back on time, sender will resend group of frames, from S to S_L. The receiver is only "loyalty" to the first incoming frame, no matter what are the conditions of the following frame, even they are correct, "ignore" them still.

Go-Back-N ARQ, normal operation

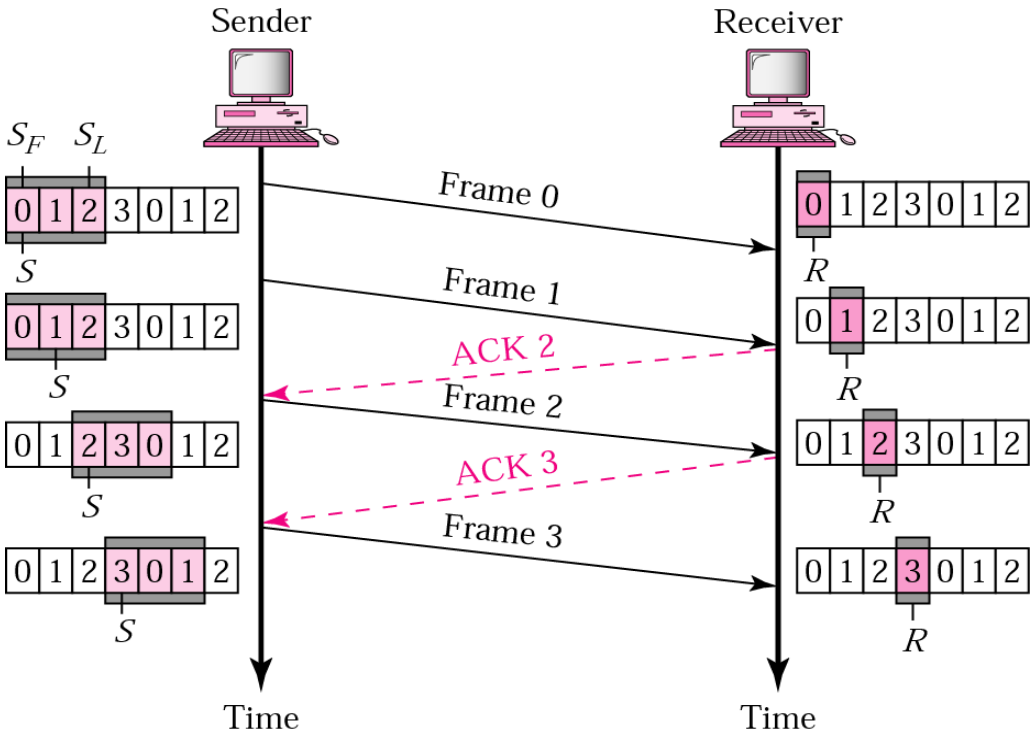

Figure 9. Go-Back-N ARQ normal operation [1]

Frame 0 & 1 send, ACK 1 & 2 back to sender. Frame 2 send, ACK 3 send back. Then frame 3 send to receiver.

Go-Back N ARQ, lost frame

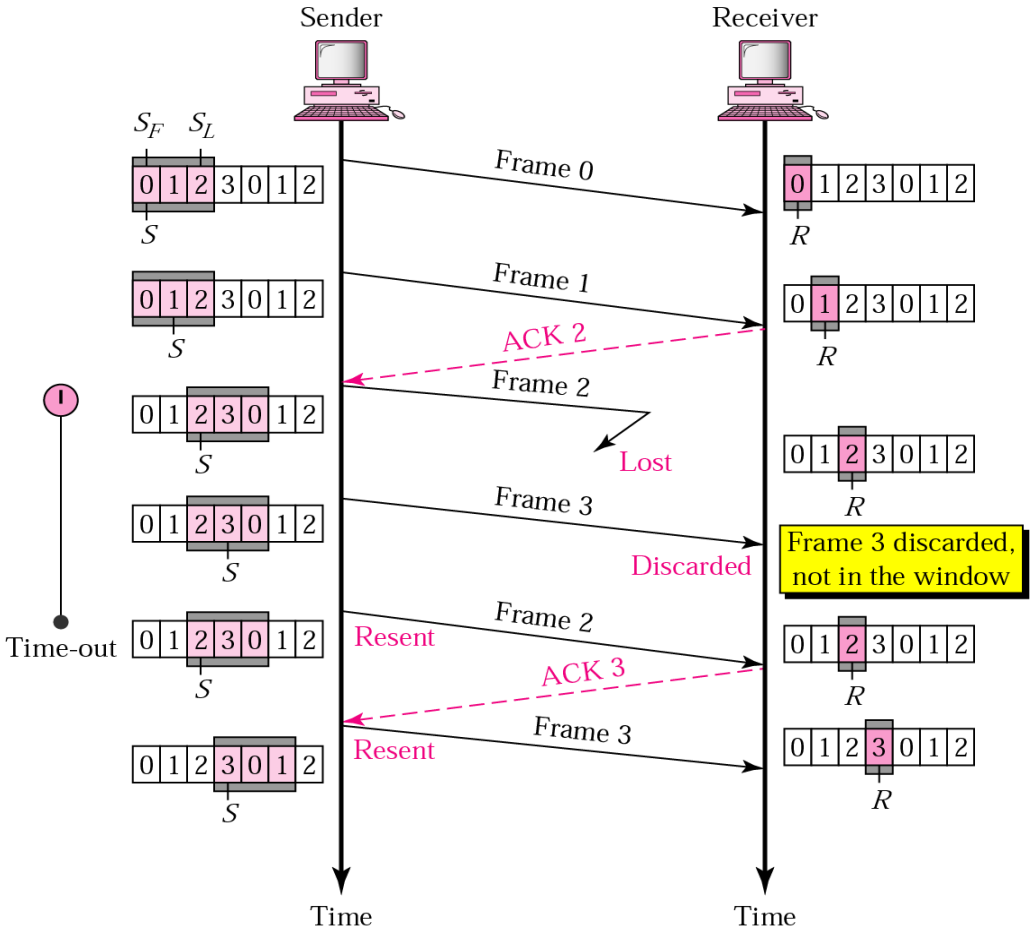

Figure 10. Go-Back-N ARQ lost frame [1]

Frame 0 & 1 send, ACK 1 & 2 back to sender. Frame 2 & 3 send, but frame 2 lost in the transmission. When frame 3 received out of order, this frame 3 will be discarded by receiver. After time out, frame 2 resent, then receiver send ACK 3 back and then frame 3 resent.

Go-Back-N ARQ, sender window size

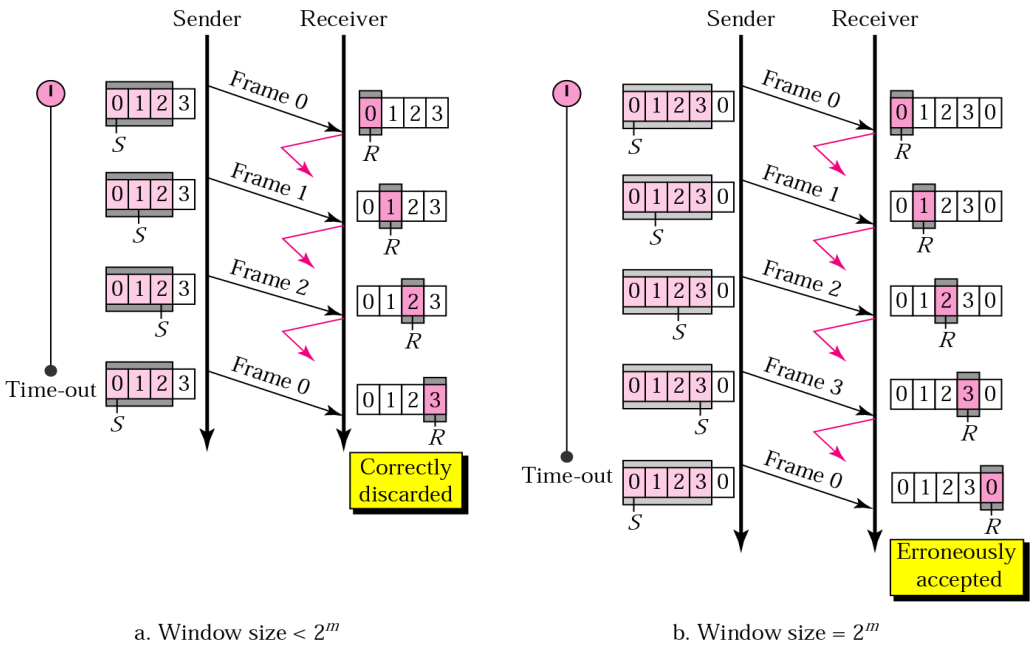

Figure 11. Go-Back-N ARQ sender window size [1]

According to the provision,

$$W = S_F + S_L + 1 = 2^m - 1$$

In figure 11 a, sender slide window size is less than 2^m. ACK 1,2,3 are lost. After frame 0 timeout, frame 0, 1, 2 are resent. But receiver is excepting frame 3 now, so when the first resent data frame 0 arrived, there are mismatched. So is could discarded correctly.

In the figure 11 b, the sender slide window size is equal to 2^m. All ACK lost, after timeout, the first frame 0 resend. At the same time, receiver slide window is point at the second frame 0. So there are “matched”, but within different cycle, the data will be erroneously accepted.

2.5 Selective Repeat ARQ

In reality, links usually noisy. Lost or damaged frame occurred very often. This means lots of frames will be discard by receiver automatically, even they are correct only with wrong order. The sender has to send those "innocent" frames again and again. Bandwidth has been used up, transmission speed slow down. All of this, made Go-Back-N ARQ is still not very high efficiency. Therefore the advanced version protocol came out. Not need to resend N frames. Instead, only the lost or damaged frame resend, this mechanism is called Selective Repeat ARQ.

Sender slide window

The control variables in Selective Repeat ARQ are same as in Go-Back-N ARQ: S_F, S_L and S. But the slide window size changed into 2^{m-1}.

Receiver slide window

Receiver has 2 control variables, R_F and R_L. The slide window size also changed into 2^{m-1}.

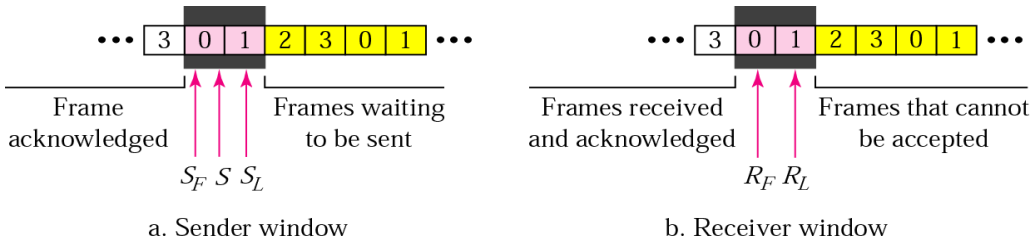

Figure 12. Selective Repeat ARQ receiver slide window [1]

Now, the sender slide window size is same as receiver's. In the Go-Back-N ARQ, each time receiver search one specific sequence number; in Selective Repeat

ARQ, each time receiver look for a range specific sequence number, from R_F to R_L.

NAK

NAK means negative acknowledgment, it does only exist in Selective Repeat ARQ.

Selective Repeat ARQ, lost frame

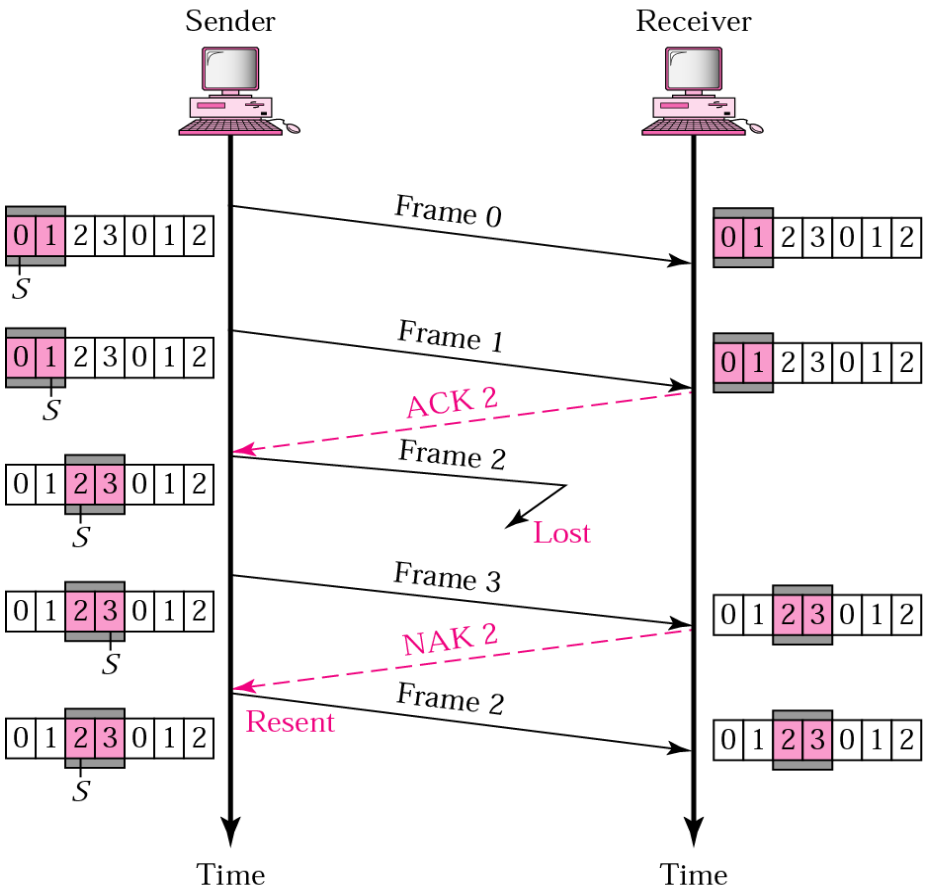

Figure 13. Selective Repeat ARQ lost frame [1]

Selective Repeat ARQ, sender window size

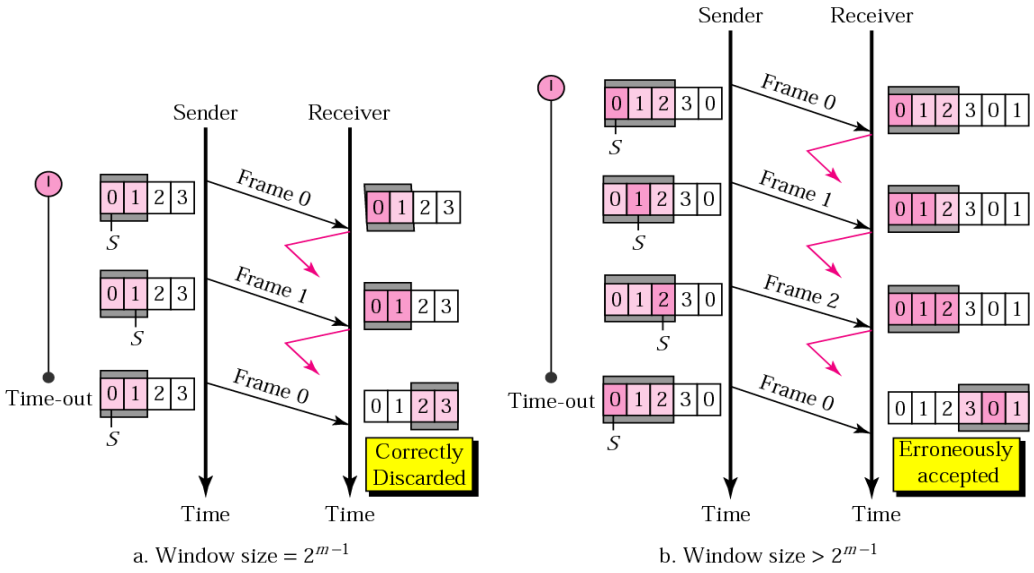

Figure 14. Selective Repeat ARQ sender window size [1]

Damaged or lost ACK NAK could be happen in the reality. But in my courseware design, assume they are always perfect received by sender. In the more complex situation, both sides of the transmission can be sender and receiver, this is so called bidirectional transmission, and this will not be discuss in this thesis.

3 JAVA SOFTWARE DESIGN

3.1 Use case

The use case of this java applet is not complex. Only one user, start from select one protocol among three, and then set the configuration of the selected protocol, the program will automatically calculate the parameters that the user chose. Start to simulate the data transfer, increase or decrease the demonstration speed. Showing the results. Reset the program, change the configuration, or back to menu, choose the other protocol.

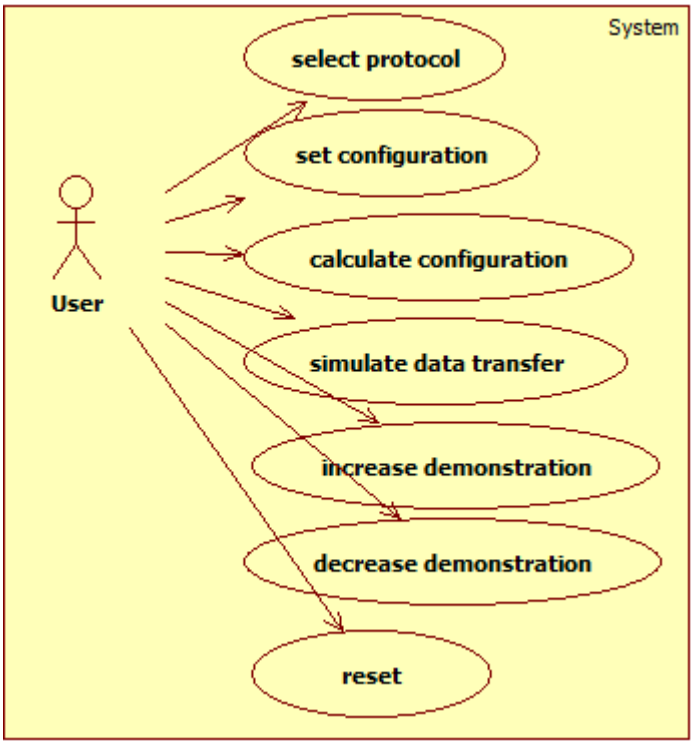

Figure 15. Use case

First of all, in my java applet is the protocol choosing menu.

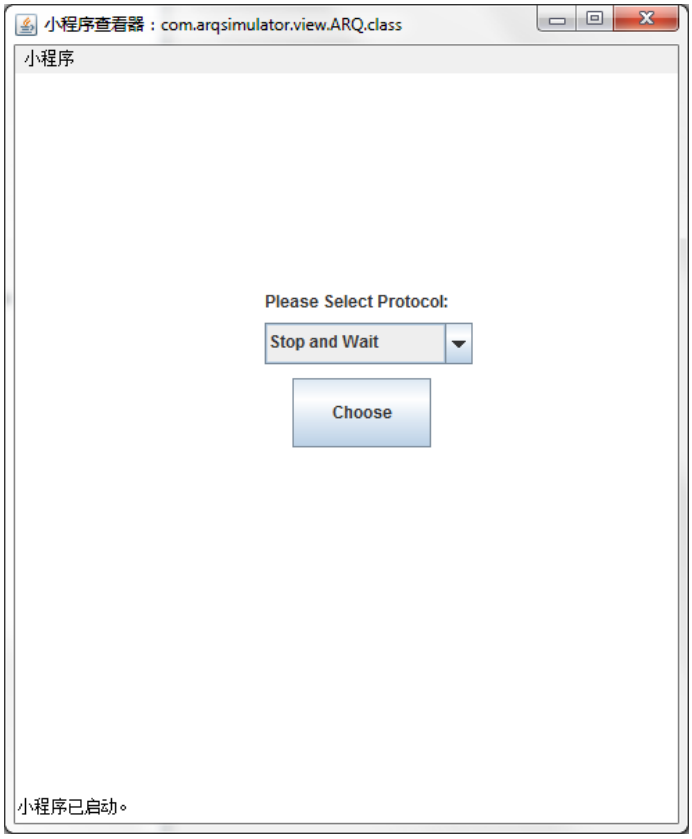

Figure 16. Protocol choosing menu 1

You can choose one of these three protocols freely.

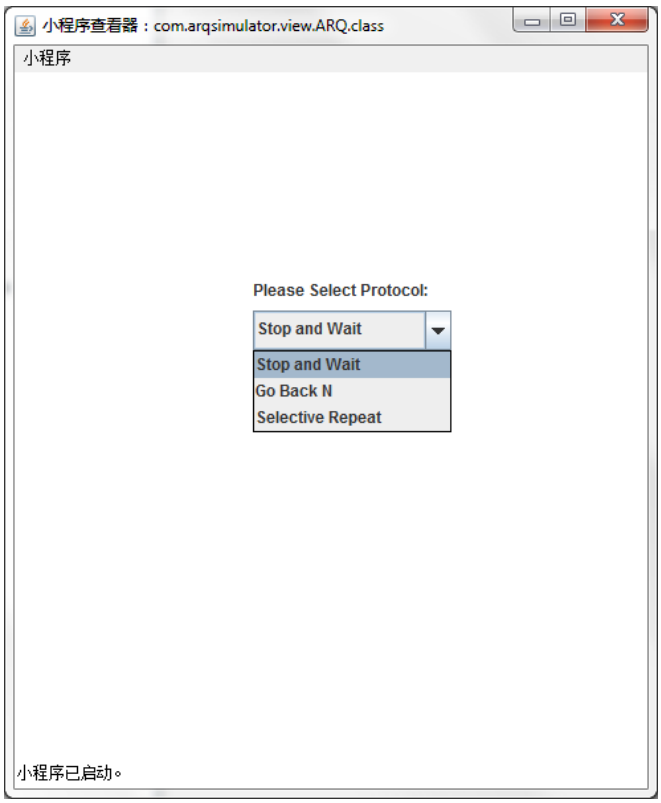

Figure 17. Protocol choosing menu 2

After we choose “Stop and Wait” and click “Choose” button, then you will see this user interface. On the left side of this user interface are all the parameters need to be used in this stop and wait protocol. The numbers inside the brackets are the range of each parameter. The numbers inside the rectangular boxes are the default value of parameters. The value below the “Frame error rate:” is the result which calculate by those parameters’ value above. Each time you change the value (must be within the range), then press the “Set” button, the program will calculate the result with three decimal.

On the right side of this user interface is the demonstration area. You can see four buttons “Fast” “Slow” “Reset” and “Send”. Below these button there is a timer, because in the stop and wait protocol, each time only one packet is transmit, so only one timer in demonstration area. Within this area, left side column rectangles are the “sender” and right side is the “receiver”. After all parameters are set, than press “Send”, the demonstration will begin, you can always press “Fast” and “Slow” buttons to adjust demonstrate speed. The demonstration will run continually until the last packet correct received. The program will generate a report, show the record and result. And then press the “Reset” to clean the area, press “Menu” to go back the protocol choosing page.

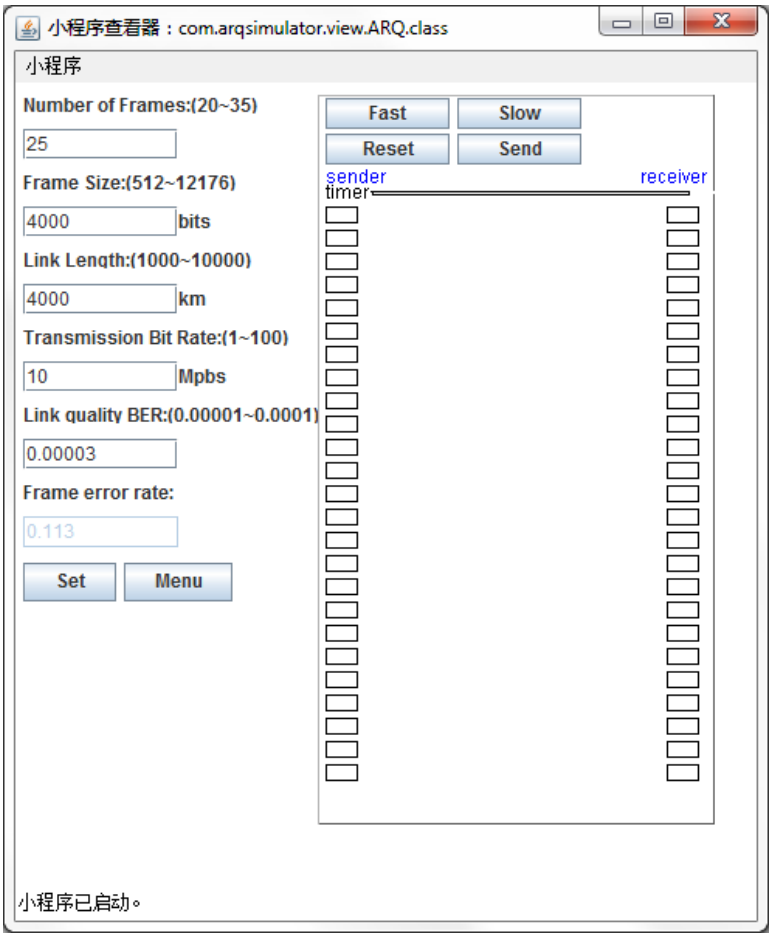

Figure 18. Stop-and-wait ARQ user interface

Go-Back-N ARQ

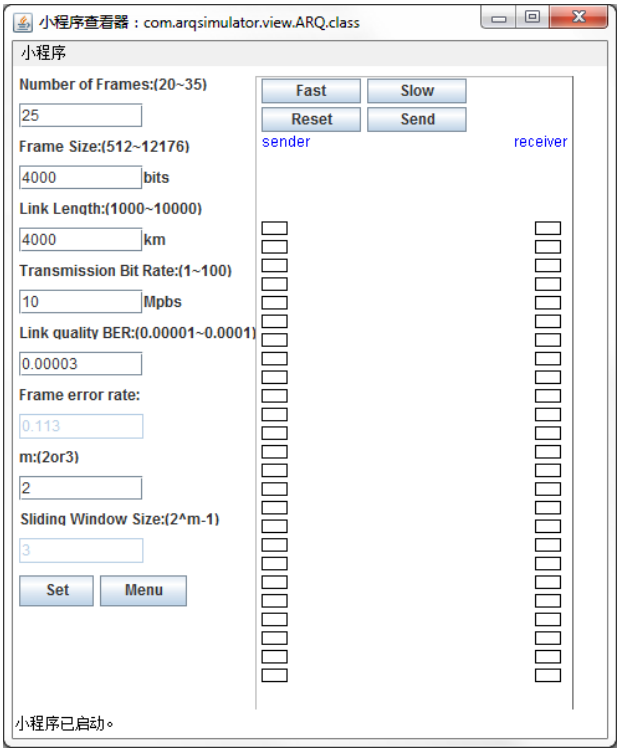

Figure 19. Go-Back-N ARQ user interface

Selective Repeat ARQ

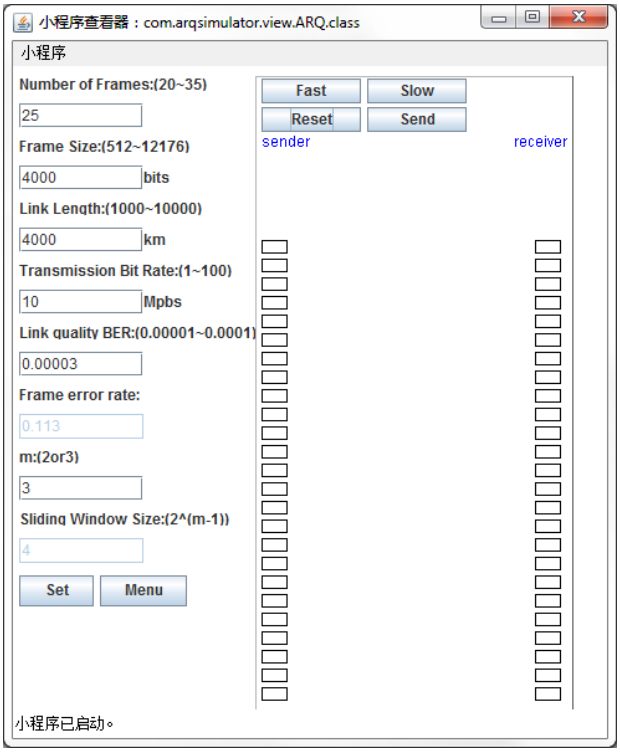

Figure 20. Selective Repeat ARQ user interface

Compare to the Stop-and-wait protocol, Go-Back-N and Selective Repeat protocols have more parameters about “Sliding window size” and “m”, the difference between these two protocols is the calculation formula.

3.2 Data structure

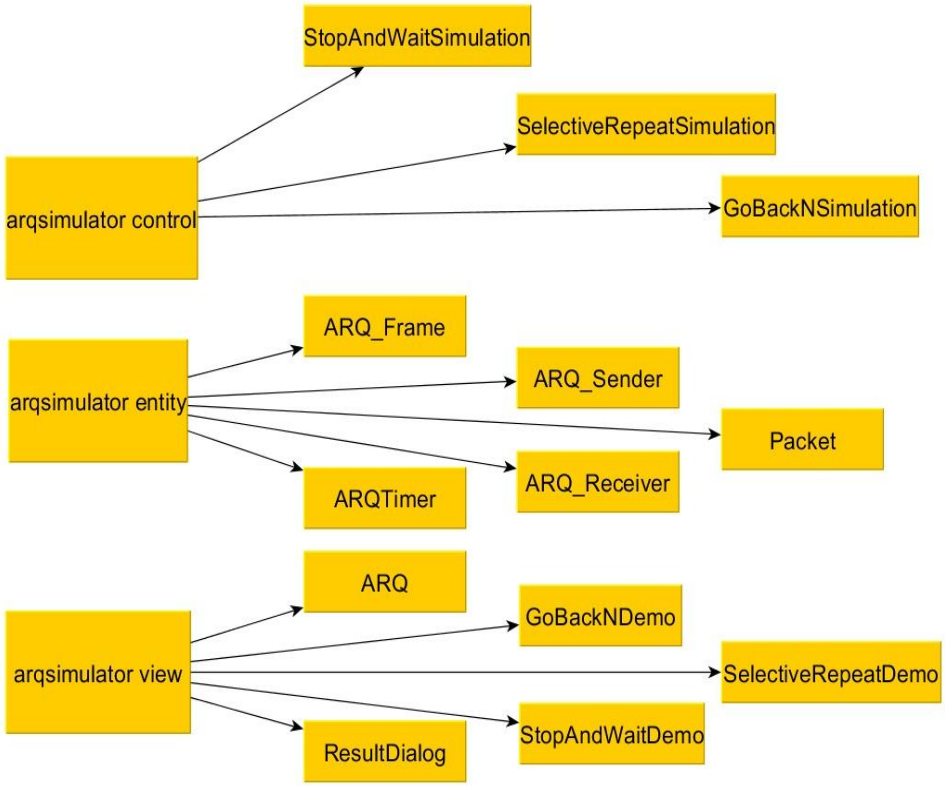

Figure 21. Data structure

The entire courseware program is composed of three parts. “arqsimulator control”, “arqsimulator entity” and “arqsimulator view”. They are planned originally representing ARQ frames transmission, simulator entity and user interface respectively. However, I modify the program very often, so the structures of the codes are confused a little bit.

In the “arqsimulator view”, there are five parts, “ARQ”, “StopAndWaitDemo”, “GoBackNDemo”, “SelectiveRepeatDemo”, and “ResultDialog”. “ARQ” is drawing the first menu page of the user interface. The drop-down protocol select. “StopAndWaitDemo” drawing the default view of the Stop-and-wait protocol. Including those buttons: “Send”, “Reset”, “Fast”, “Slow”. And marked those

parameters' selectable range. If the user fills the parameters out of range, it will show the warning to tell the proper range. Calculate with those parameters. "GoBackNDemo" have the same function to the Go-Back-N protocol; "SelectiveRepeatDemo" does the same thing to the Selective Repeat protocol. Actually there three parts could be combining together. "ResultDialog" will show the data record of the simulation, and the result according to the data and calculation.

In the "arqsimulator entity", there are five parts: "ARQ_Frame", "ARQ_Receiver", "ARQ_Sender", "ARQTimer" and "Packet". "ARQ_Frame" is drawing the basic frames in the simulation. "Packet" show shows the frame in movement. When the frame transmits to receiver, the color is yellow. If it's ACK back, the color is red, if its NAK back, the color is black. "ARQ_Sender" drawing sender side column, how many rectangular should be there according to the parameter and the sender slider window size. "ARQ_Receiver" do the same thing to the receiver side. "ARQTimer" is responds for the timer, each frame have their own individual timer.

In the "arqsimulator control", there are three parts. "StopAndWaitSimulation" is responds for Stop-and-Wait ARQ frames transmission simulation. It counts frame number and slide window number. After "ARQTimer" timeout, retransmit the frame. If it receives ACK, than transmit next frame. For each frame, there is a random number will be generated between 0 and 1, compare this number to the calculated frame error rate, if it smaller, then this frame will be dropped. "GoBackNSimulation" and "SelectiveRepeat" did the same thing to Go-Back-N and Selective Repeat protocol.

4 TESTING AND CALCULATION

Test the stop and wait ARQ. Fill the parameters, 25 frames, frame size 4000 bits, link length 4000 km, transmission bit rate 10 Mbps, link quality BER is 0.00003.

According to the formula

$$FER = 1 - (1 - BER)^{frames\ size}$$

$$FER = 1 - (1 - 0.00003)^{4000} \approx 0.113$$

So, the program gets the right result of frame error rate.

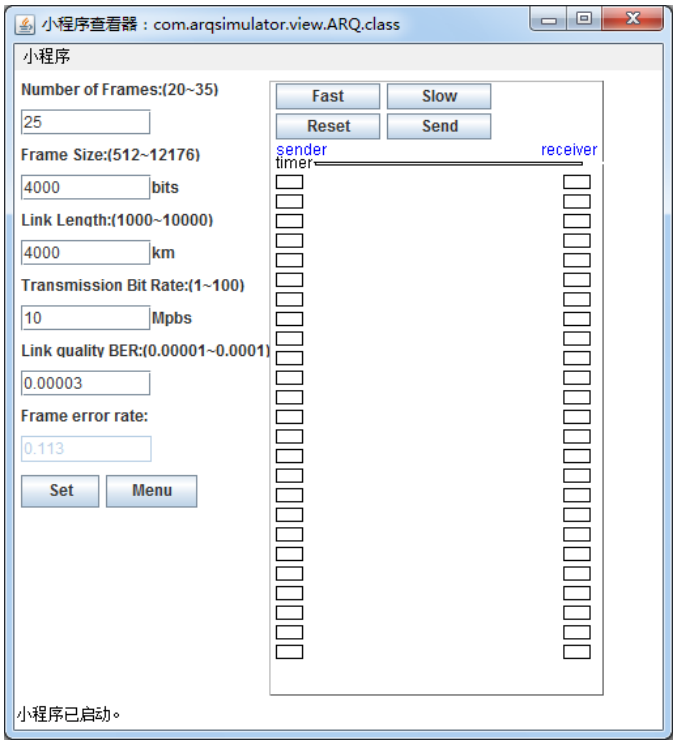

Figure 22. Stop-and-wait ARQ parameter calculation

Start the simulation, nothing wrong with the progress.

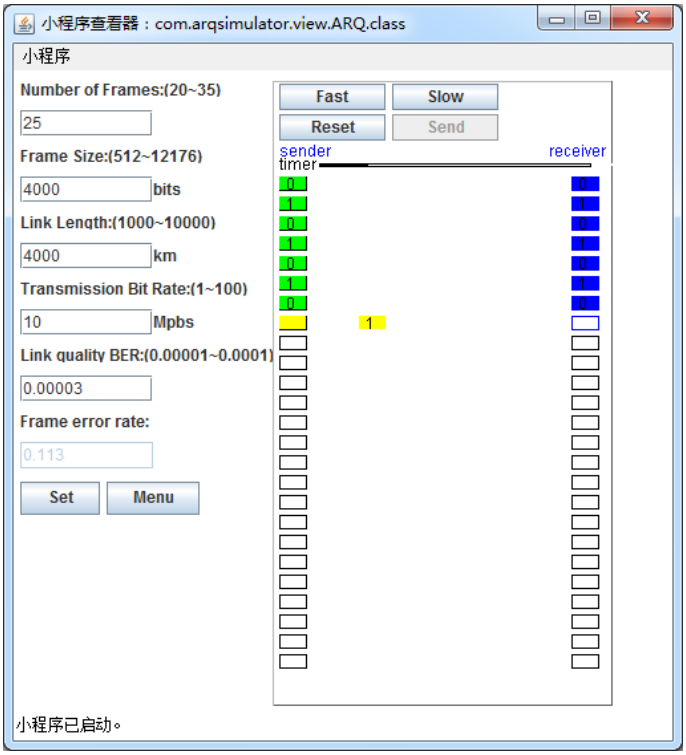

Figure 23. Stop-and-wait ARQ in progress

After the simulation finished, the result dialog shows up, it says 3 frames are randomly error, and they have been retransmitted. In all stop and wait ARQ, there is no automatically discard frames, so it's 0. Theoretical total delivery time is 0.7629s, and simulation total delivery time is 0.7578s.

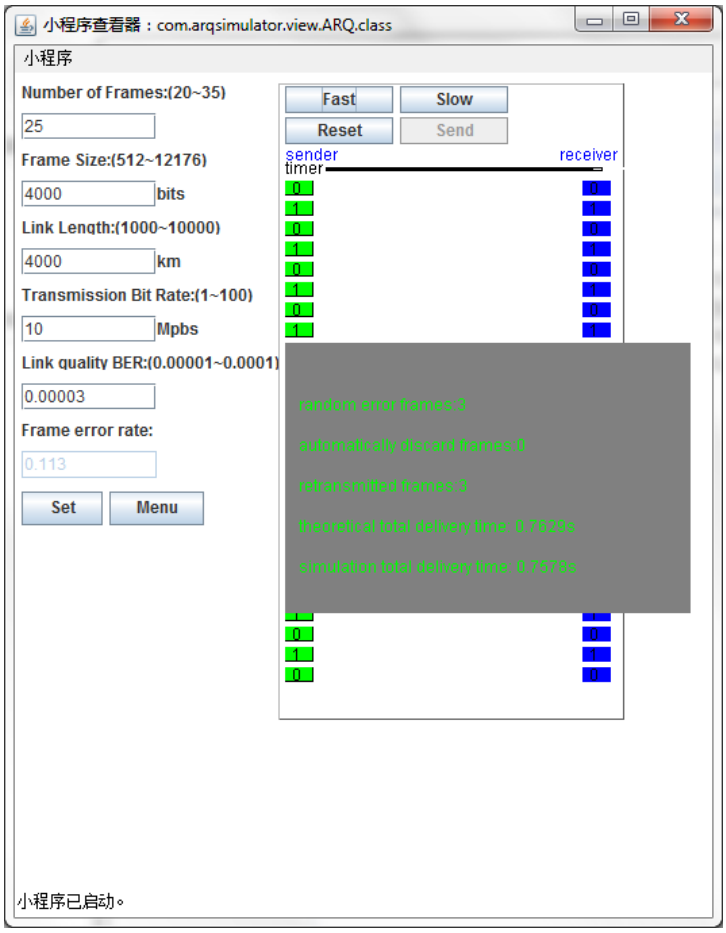

Figure 24. Stop-and-wait ARQ result

According to the formula:

Theoretical total delivery time:

$$\left(\frac{frame\ size}{transmission\ bit\ rate} + \frac{2 * link\ length}{speed\ of\ light}\right) * \frac{number\ of\ frames}{1-FER}$$

Theoretical total delivery time is

(4000/10000000+2*4000000/300000000)*(25/ (1-0.113)) = 0.7629 seconds, in 4 decimal

Compare to the simulation total delivery time:

$$\left(\frac{frame\ size}{transmission\ bit\ rate} + \frac{2 * link\ length}{speed\ of\ light}\right)$$*(number of frames +error frames)

(4000/10000000+2*4000000/300000000)*(25+3) = 0.7578 seconds, in 4 decimal

So the result is accepted.

Go-Back-N ARQ, we fill the same parameters, 25 frames, frame size 4000 bits, link length 4000km, transmission bit rate is 10 Mbps, link quality BER is 0.00003, and slide window size is 2.

According to the formula:

$$windows\ size = 2^m - 1 = 2^2 - 1 = 3$$

$$FER = 1 - (1 - BER)^{frames\ size} = 1 - (1 - 0.00003)^{4000} \approx 0.113$$

So, the program gets the right result so far.

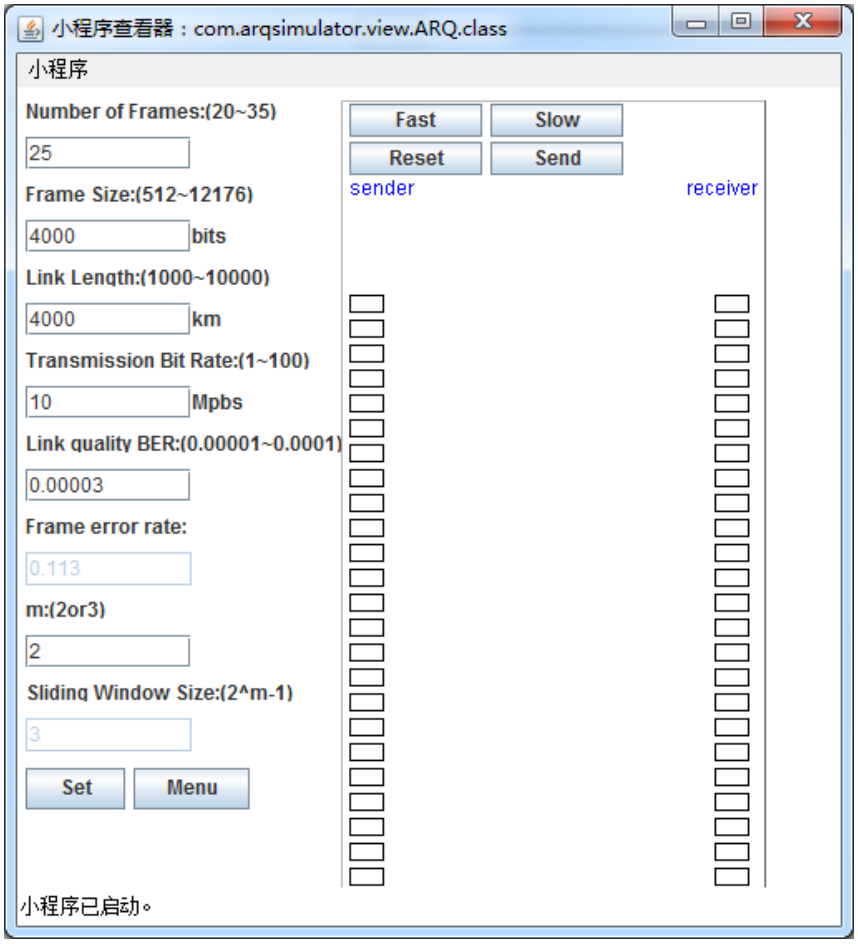

Figure 25. Go-Back-N ARQ parameter calculation

Start the simulation, Go-back-N's progress is ok.

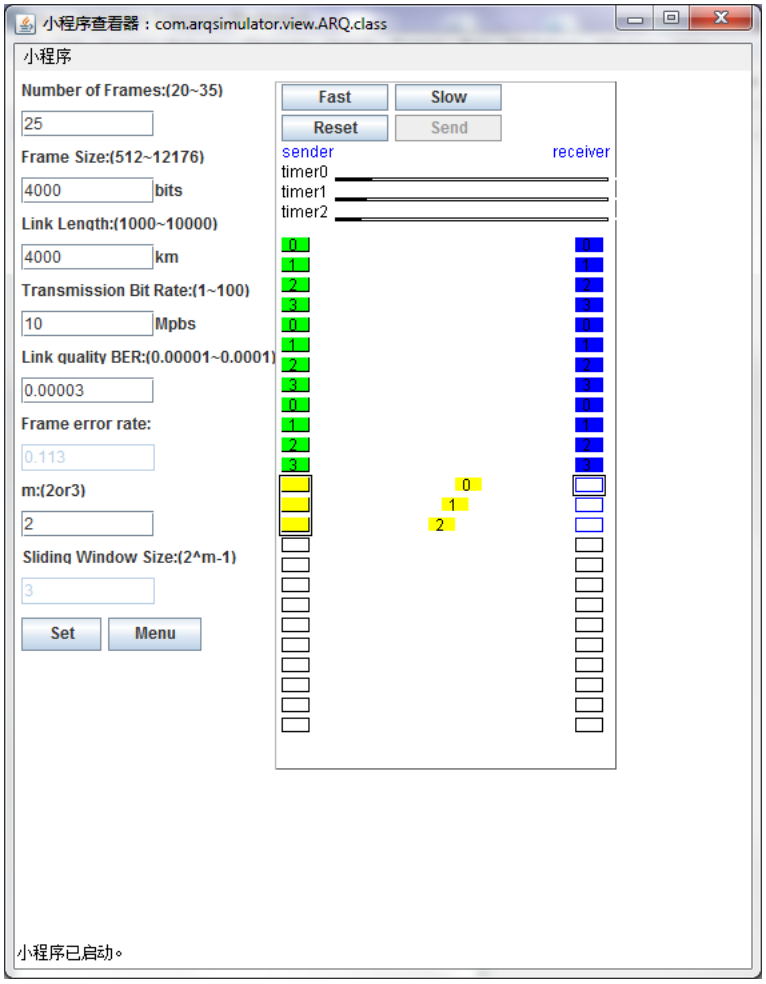

Figure 26. Go-Back-N ARQ in progress

The result shows there are 3 random error frames, and 6 frames are automatically discarded because those 3 error frames, and totally 9 frames retransmitted.

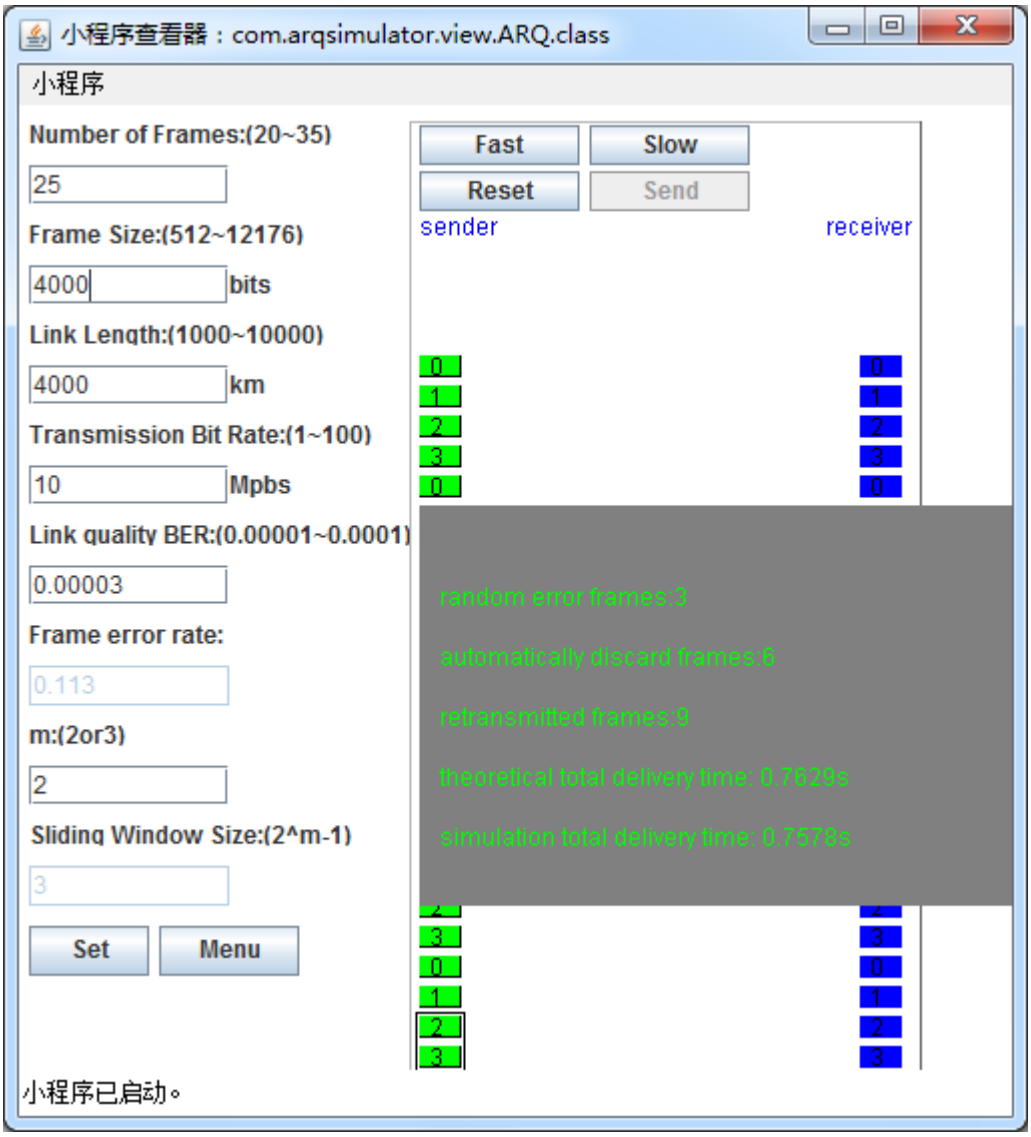

Figure 27. Go-Back-N ARQ result

According to the formula, the theoretical total delivery time and simulation total delivery time are quiet closed, so the result is acceptable. However, about 20% when I run the go back N ARQ, there are some random error occurs, especially when I increase the speed of the simulation. So there is a bug, which I cannot fix

Selective repeat ARQ

Fill the same parameters, but give the m value is 3. Frame error rate is still 0.113.

But $slide\ window\ size = 2^{m-1} = 2^{3-1} = 4$.

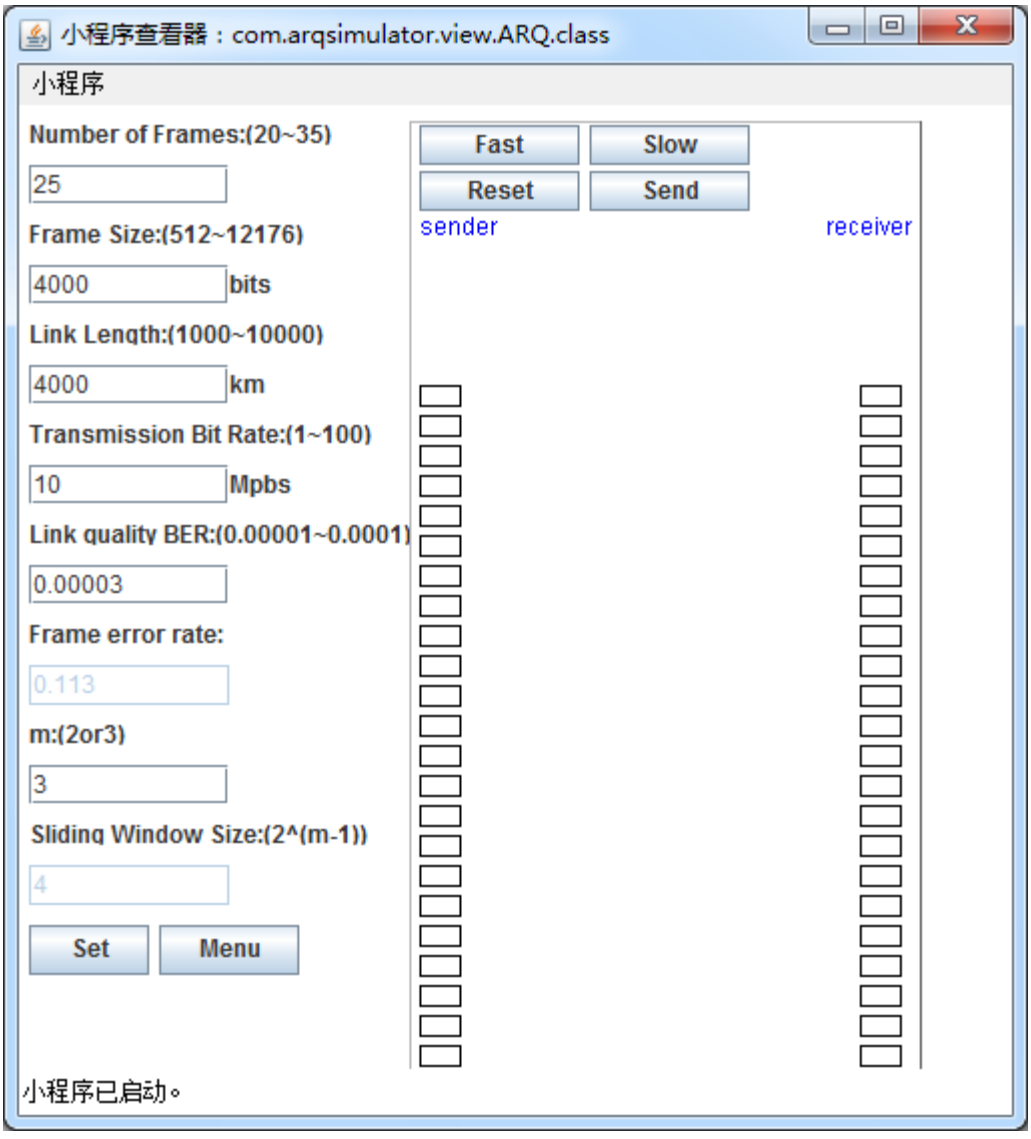

Figure 28. Selective Repeat ARQ parameter calculation

Run the selective repeat ARQ simulation. Progress is ok.

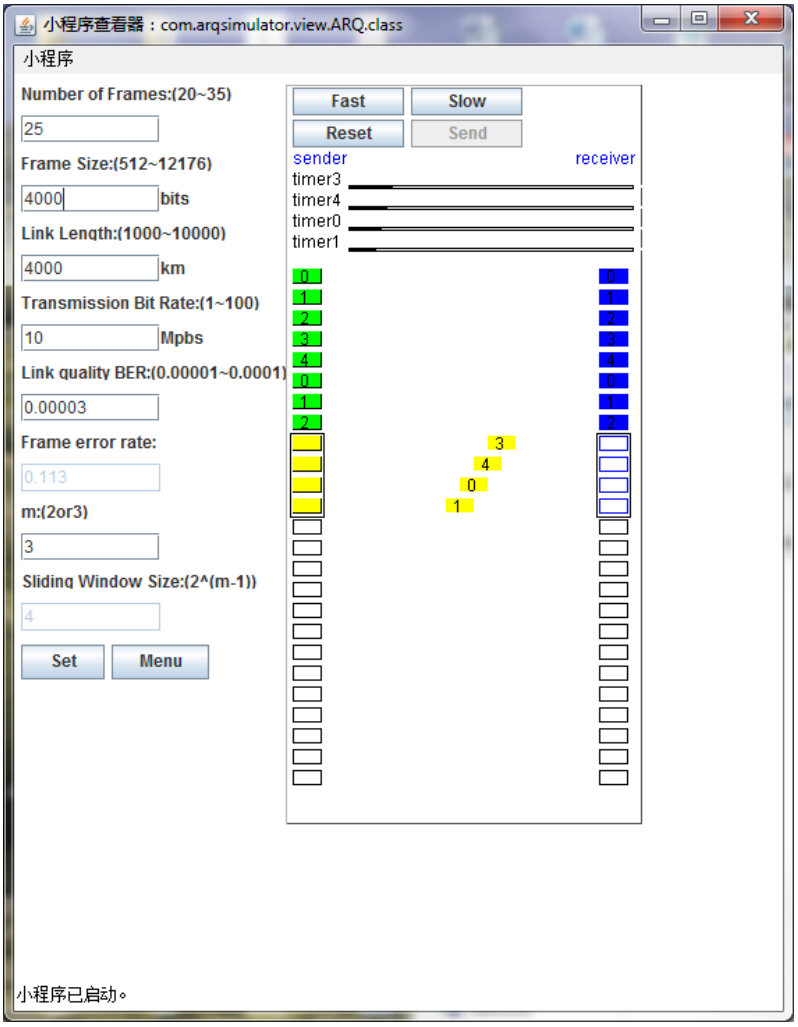

Figure 29. Selective Repeat ARQ in progress

The result is 2 random error frames and 2 frames retransmitted.

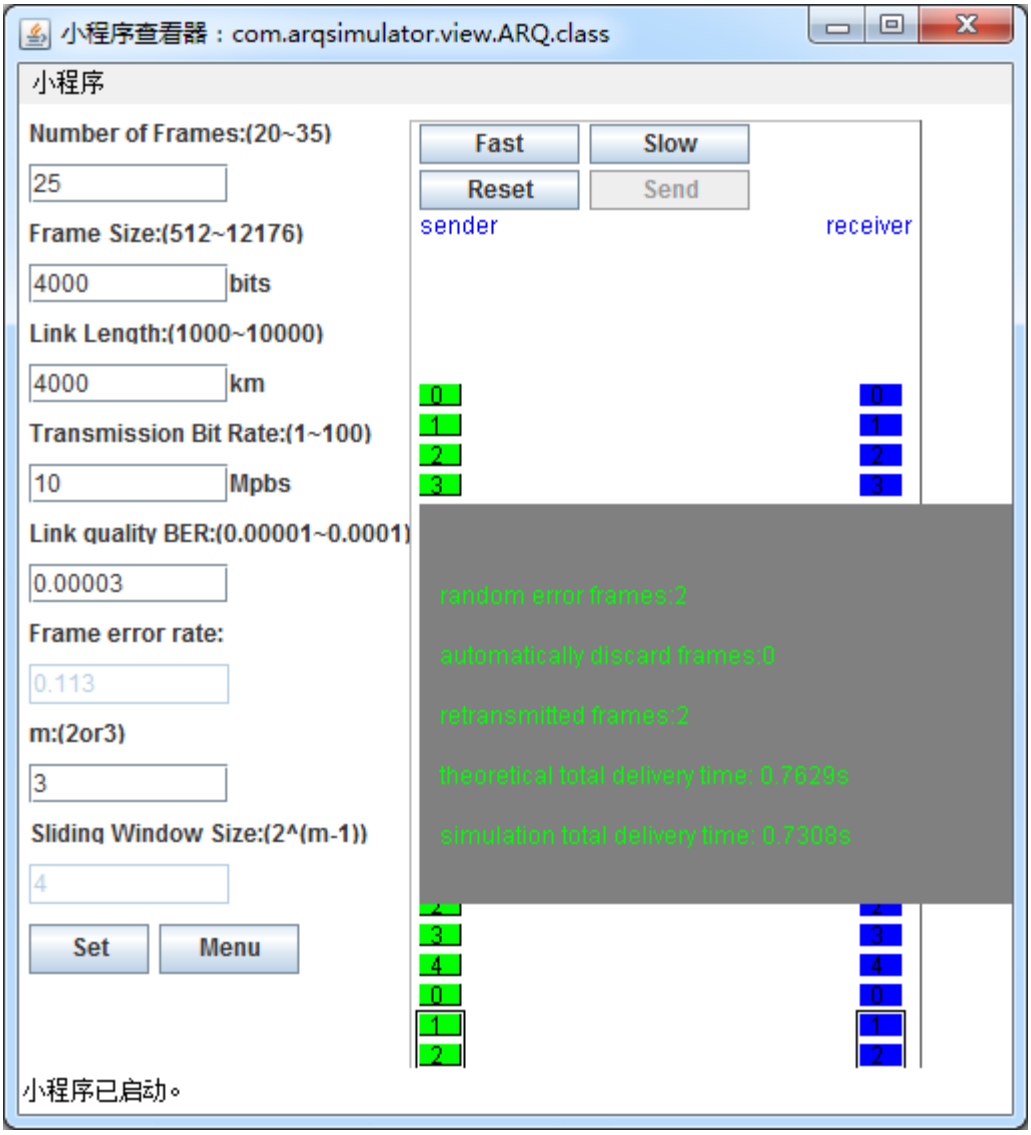

Figure 30. Selective Repeat ARQ result

Compare simulation and theory, result are closed. But in this selective repeat ARQ simulation about 30% run times have bugs.

5 SUMMARY AND CONCLUSION

Comparing three different ARQ protocols, I get this form showing below.

	Stop-and-Wait	Go-back-N	Selective Repeat
Sender slide window size	1	2^m-1	2^{m-1}
Receiver slide window size	1	1	2^{m-1}
ACK	Yes	Yes	Yes
NAK	No	No	Yes
Frame sequence	0,1,0,1...	0...2^m-1	0...2^m-1
Bandwidth utilization	Low	Medium	high

Figure 31. Three ARQ protocols comparing table

This is not just simply to say which protocol is better or faster than the others. It depends on the environment and all those parameters. Choose the proper protocol for suitable situation. This ARQ simulation courseware is not perfect running, but basically those simulation results are meet the theory. When I programming and writing this thesis, I have lots of trouble, the more I dig into this ARQ theory, the more difficult I get, lots of detail things are completely different from what I though earlier. And because my major is the telecommunication, so the java applet took me long time to handle. Thanks to my supervisor Mr. Gao Chao's help. I finally finish this.

REFERENCES

1. Behrouz A. Forouzan, “Data Communications and Networking”, (third edition), published by McGraw-Hill Education, 2004.

2. FER reference from Aftab Ahmad, “Data Communication Principles: For Fixed and Wireless Networks”, published by Kluwer Academic Publishers, 2003, page 143 to 145.